python-basic-4

October 20, 2024

[10]:
```python
# Start with Basic --
```

[12]:
```python
print("hello World!")
```

```
hello World!
```

[14]:
```python
# with variable:
x = 4
y = 6
print(x+y)
```

```
10
```

[20]:
```python
# Swapping the value:
# 1st:
a = 4
b = 6
a = a+b
b = a-b
a = a-b
print("a:",a)
print("b:",b)
```

```
a: 6
b: 4
```

[21]:
```python
# 2nd: without 3rd variable:
a = 4
b = 6
a,b = b,a
print("a:",a)
print("b:",b)
```

```
a: 6
b: 4
```

[18]:
```python
# User Input : Str
x = input("Type something:")
```

```
print(x)
```

```
Type something: Hi my name is Manish Bisht.
```

```
Hi my name is Manish Bisht.
```

[16]:
```
# User Input: int
x = int(input("Enter a number:"))
print(x)
```

```
Enter a number: 12
```

```
12
```

[15]:
```
# IF Else condition:
x = 4
if x>10:
    print("Greater")
else:
    print("less")
```

```
less
```

[24]:
```
a = 4
b = 4
if a > 5:
    print("Greater")
elif a == b:
    print("Equal")
else:
    print("less")
```

```
Equal
```

0.1 Functions

[2]:
```
# split a string:
a = "Honesty is the best policy"
print(a.split()[3])
```

```
best
```

[3]:
```
# Rsplit:
a = "Honesty is the best policy"
print(a.rsplit("s",(2)))
```

```
['Honesty i', ' the be', 't policy']
```

0.2 For Loop:

[5]:
```
# PRint 1 to 10:
for i in range(1,11):
    print(i)
```

```
1
2
3
4
5
6
7
8
9
10
```

[6]:
```
for i in range(1,11):
    print(i,end=" ")
```

```
1 2 3 4 5 6 7 8 9 10
```

[7]:
```
# Now to print counting 10 to 1:
for i in range(10,0,-1):
    print(i,end=" ")
```

```
10 9 8 7 6 5 4 3 2 1
```

[8]:
```
# Mathematic table:
n =  int(input("Enter a number:"))
for i in range(n,n*10+1,n):
    print(i,end=" ")
```

```
Enter a number: 5

5 10 15 20 25 30 35 40 45 50
```

[9]:
```
# in a better way:
n = 6
for i in range(1,11):
    print(n,"*",i,"=",n*i)
```

```
6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
```

```
6 * 8 = 48
6 * 9 = 54
6 * 10 = 60
```

[10]:
```
sm = 0
for i in range(1,11):
    sm = sm+i
print(sm)
```

```
55
```

[11]:
```
# Factorial Program:
n = 5
sm = 1
for i in range(1,n+1):
    sm = sm*i
print(sm)
```

```
120
```

[12]:
```
# Prime NUmber Program:
n = 7
flag = 0
for i in range(2,n//2+1):
    if n%i==0:
        flag = 1
        break
if flag == 0:
    print("Prime Number")
else:
    print("NOt Prime number")
```

```
Prime Number
```

[13]:
```
# Perfect Number:
n = 28
sm = 0
for i in range(1,n//2+1):
    if n%i==0:
        sm +=i
if sm == n:
    print("Perfect number")
else:
    print("not a perfect number")
```

```
Perfect number
```

[14]:
```
# Lcm of two numbers:
a = 4
b = 6
mx = a if a>b else b
for i in range(mx,a*b+1):
    if i%a==0 and i%b==0:
        print("LCM:",i)
        break
```

```
LCM: 12
```

[15]:
```
# HCf of two numbers:
a = 12
b = 15
mx = a if a<b else b
for i in range(2,mx+1):
    if a&i==0 and b%i==0:
        print("HCF:",i)
        break
```

```
HCF: 3
```

[16]:
```
# all prime number between 1 to 100:
for x in range(1,101):
    n = x
    flag = 0
    for i in range(2,n//2+1):
        if n%i==0:
            flag=1
            break
    if flag == 0:
        print(x,end=" ")
```

```
1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

[17]:
```
# All perfect numbers between 1 to 100:
for x in range(1,101):
    n = x
    sm = 0
    for i in range(1,n//2+1):
        if n%i==0:
            sm += i
    if sm == n:
        print(x,end=' ')
```

```
6 28
```

[7]:
```
# To create a loop with a charcter:

for i in range(65,70):
    for j in range(65,i+1):
        print(chr(i),end="")
    print()
```

```
A
BB
CCC
DDDD
EEEEE
```

[8]:
```
for i in range(65,70):
    for j in range(65,i+1):
        print(chr(j),end="")
    print()
```

```
A
AB
ABC
ABCD
ABCDE
```

[1]:
```
for i in range(1,7):
    for j in range(1,i+1):
        print(j,end="")
    print()
```

```
1
12
123
1234
12345
123456
```

[6]:
```
# Box with astricts:
for j in range(1,8):
    for i in range(1,8):
        print("*",end=" ")
    print()
```

```
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
```

[18]:
```python
# To make a triangle with astricts:
for j in range(1,7):
    for k in range(1,7-j):
        print(" ",end="")
    for i in range(1,j+1):
        print("*",end=" ")
    print()
```

```
     *
    * *
   * * *
  * * * *
 * * * * *
* * * * * *
```

[1]:
```python
# right angle triangle:
for j in range(1,7):
    for i in range(1,j+1):
        print("*",end=" ")
    print()
```

```
*
* *
* * *
* * * *
* * * * *
* * * * * *
```

[2]:
```python
# right angle triangle upside down:
for j in range(6,0,-1):
    for i in range(1,j+1):
        print("*",end=" ")
    print()
```

```
* * * * * *
* * * * *
* * * *
* * *
* *
*
```

[21]:
```python
# Upside down triangle:
for j in range(6,0,-1):
    for k in range(1,7-j):
        print(" ",end="")
    for i in range(1,j+1):
        print("*",end=" ")
    print()
```

```
* * * * * *
 * * * * *
  * * * *
   * * *
    * *
     *
```

[4]:
```
# Mirror right angle triangle:
for j in range(1,7):
    for k in range(1,7-j):
        print(" ",end=" ")
    for i in range(1,j+1):
        print("*",end=" ")
    print()
```

```
          *
        * *
      * * *
    * * * *
  * * * * *
* * * * * *
```

[3]:
```
# Mirror right angle triangle upside down:
for j in range(6,0,-1):
    for k in range(1,7-j):
        print(" ",end=" ")
    for i in range(1,j+1):
        print("*",end=" ")
    print()
```

```
* * * * * *
  * * * * *
    * * * *
      * * *
        * *
          *
```

[24]:
```
# program to make a table with loop:
for j in range(1,11):
    for i in range(1,11):
        print(i*j,end=" ")
    print()
```

```
1 2 3 4 5 6 7 8 9 10
2 4 6 8 10 12 14 16 18 20
```

```
3 6 9 12 15 18 21 24 27 30
4 8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
6 12 18 24 30 36 42 48 54 60
7 14 21 28 35 42 49 56 63 70
8 16 24 32 40 48 56 64 72 80
9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
```

[25]:
```
# WHILE LOOP:
```

[26]:
```
# print 1 to 10:
a = 1
while a<=10:
    print(a,end=" ")
    a = a+1
```

```
1 2 3 4 5 6 7 8 9 10
```

[27]:
```
# Program for reverse number:
n = 946165
sm = 0
while n>0:
    r = n%10
    n = n//10
    sm = sm*10+r
print(sm)
```

```
561649
```

[5]:
```
#  Program for armstrong number:
n = 153
x = n
l = len(str(n))
sm = 0
while n>0:
    r = n%10
    n = n//10
    sm = sm+r**l
if sm == x:
    print("Armstrong number")
else:
    print("Not a armstrong number")
```

```
Armstrong number
```

[6]:
```
# Program for palindrom:
n = 25352
```

```
x = n
sm = 0
while n>0:
    r = n%10
    n = n//10
    sm = sm*10+r
if sm == x:
    print("Palindrom number")
else:
    print("not a palindrom number")
```

```
Palindrom number
```

[30]:
```
# Program to find all the armstrong number:
for i in range(1,10000):
    n = i
    l = len(str(n))
    sm = 0
    while n>0:
        r = n%10
        n = n//10
        sm = sm+r**l
    if sm == i:
        print(i,end=' ')
```

```
1 2 3 4 5 6 7 8 9 153 370 371 407 1634 8208 9474
```

[31]:
```
# Program to find all the palindrome numbers:
for i in range(100,1001):
    n = i
    sm = 0
    while n>0:
        r = n%10
        n = n//10
        sm = sm*10+r
    if sm == i:
        print(i,end=" ")
```

```
101 111 121 131 141 151 161 171 181 191 202 212 222 232 242 252 262 272 282 292
303 313 323 333 343 353 363 373 383 393 404 414 424 434 444 454 464 474 484 494
505 515 525 535 545 555 565 575 585 595 606 616 626 636 646 656 666 676 686 696
707 717 727 737 747 757 767 777 787 797 808 818 828 838 848 858 868 878 888 898
909 919 929 939 949 959 969 979 989 999
```

[32]:
```
# Program to find all the prime palindrome numbers:
for i in range(100,1001):
    n = i
    sm = 0
    while n>0:
        r = n%10
        n = n//10
        sm = sm*10+r
    if sm == i:
        flag = 0
        for j in range(2,i//2+1):
            if i%j == 0:
                flag = 1
                break
        if flag == 0:
            print(i,end=' ')
```

```
101 131 151 181 191 313 353 373 383 727 757 787 797 919 929
```

[33]:
```
# Find All the armstrong prime numbers:
for i in range(1,10000):
    n = i
    l = len(str(n))
    sm = 0
    while n>0:
        r = n%10
        n = n//10
        sm = sm+r**l
    if sm == i:
        flag = 0
        for j in range(2,i//2+1):
            if i%j == 0:
                flag = 1
                break
        if flag == 0:
            print(i,end=' ')
```

```
1 2 3 5 7
```

[34]:
```
# Program to find the the number is Prime number or not:
n = 10
i = 2
while i * i <= n:
    if n % i == 0:
        print(n, "is not a prime number")
        break
    i += 1
else:
    print(n, "is a prime number")
```

```
10 is not a prime number
```

[35]:
```
# Find all the perfect number only using while loop:
i = 1
while i<=100:
    j = 1
    sm = 0
    while j<=i//2:
        if i%j==0:
            sm=sm+j
        j=j+1
    if i==sm:
        print(sm)
    i=i+1
```

```
6
28
```

[36]:
```
# Finding LCM:
a = 12
b = 15
mx = a if a>b else b
i = mx
while i<=a*b:
    if i%a==0 and i%b==0:
        print("LCM is:",i)
        break
    i = i+1
```

```
LCM is: 60
```

[12]:
```
# Collections : String, List, Tuple, Dictionary, Set
# String: Collection of character enclosed with quotations
# Collection:
# Arrays::collection of similar type of values
```

[10]:
```
# with user input: program to make a small name:
a = input("enter name:")
a=" "+a

for i in range(0,len(a)):
    if a[i] == " ":
        print(a[i+1].upper(),end=".")
```

```
enter name: sachin ramesh tendulkar
```

```
S.R.T.
```

[11]:
```
# Qns: like M.K.Gandhi:
```

```
n = input("Enter The Name:")
j = n.split()
x = " "
for i in j[:-1]:
    x = x + i[0].upper() + "."
x = x+j[-1].capitalize()
print(x)
```

```
Enter The Name: sachin ramesh tendulkar

 S.R.Tendulkar
```

[38]:
```
# LISt:
data = [12,56,4,456,46,156,56]
print(data)

# Slicing operators:
print(data[0:7])
print(data[2:5])
print(data[-1::-1])
print(data[:-1])
print(data[2:])
```

```
[12, 56, 4, 456, 46, 156, 56]
[12, 56, 4, 456, 46, 156, 56]
[4, 456, 46]
[56, 156, 46, 456, 4, 56, 12]
[12, 56, 4, 456, 46, 156]
[4, 456, 46, 156, 56]
```

[39]:
```
# travarsal operations:
data = [12,56,4,456,46,156,56]
for i in data:
    print(i,end=" ")
```

```
12 56 4 456 46 156 56
```

[40]:
```
data = [12,56,4,456,46,156,56]
for i in range(0,len(data)):
    print(data[i],end=" ")
```

```
12 56 4 456 46 156 56
```

[41]:
```
data = [12,56,4,456,46,156,56]
for i in data:
    print(data)
```

```
[12, 56, 4, 456, 46, 156, 56]
[12, 56, 4, 456, 46, 156, 56]
```

```
[12, 56, 4, 456, 46, 156, 56]
[12, 56, 4, 456, 46, 156, 56]
[12, 56, 4, 456, 46, 156, 56]
[12, 56, 4, 456, 46, 156, 56]
[12, 56, 4, 456, 46, 156, 56]
```

[42]:
```
# Reverse for of data:
data = [12,56,4,456,46,156,56]
for i in range(len(data)-1,-1,-1):
    print(data[i],end=" ")
```

```
56 156 46 456 4 56 12
```

[43]:
```
# Some Functions:
data = [12,56,4,456,46,156,56]
print(min(data))
print(max(data))
print(len(data))
print(sum(data))
print(data.count(4))
```

```
4
456
7
786
1
```

[44]:
```
# Add data:
# To add at last:
data = [12,56,4,456,46,156,56]
data.append(23)
print(data)
# To print data at a specified placr:
data.insert(4,61)
print(data)
```

```
[12, 56, 4, 456, 46, 156, 56, 23]
[12, 56, 4, 456, 61, 46, 156, 56, 23]
```

[45]:
```
# To make chnages is list:
data = [12,56,4,456,46,156,56]
data[2]=45
print(data)
# To make multiple changes:
data[2:4] = 25,45
print(data)

data[2:7:2] = 12,45,56
```

```
print("New data:",data)
```

```
[12, 56, 45, 456, 46, 156, 56]
[12, 56, 25, 45, 46, 156, 56]
New data: [12, 56, 12, 45, 45, 156, 56]
```

[46]:
```
# Delete
# to delete the values inside the list:
data = [12,56,4,456,46,156,56]
del data[2]
print(data)
# to delete all the values:
data.clear()
print(data)
```

```
[12, 56, 456, 46, 156, 56]
[]
```

[47]:
```
# To delete last value:
data = [12,56,4,456,46,156,56]
data.pop()
print(data)
```

```
[12, 56, 4, 456, 46, 156]
```

[48]:
```
# To print the last value:
data = [12,56,4,456,46,156,56]
x = data.pop()
print(x)
```

```
56
```

[49]:
```
# To remove a perticuler Value:
data = [12,56,4,456,46,156,56]
data.remove(456)
print(data)
# To delete value using index:
data.pop(4)
print(data)
```

```
[12, 56, 4, 46, 156, 56]
[12, 56, 4, 46, 56]
```

[50]:
```
# Sort
```

[51]:
```
# Sort in accending order:
data = [12,56,4,456,46,156,56]
data.sort()
```

```
print(data)
```

```
[4, 12, 46, 56, 56, 156, 456]
```

[52]:
```
# Sort in decending order:
data = [12,56,4,456,46,156,56]
data.sort()
data.reverse()
print(data)
```

```
[456, 156, 56, 56, 46, 12, 4]
```

[53]:
```
# user input list:
data = []
n = int(input("Enter limit:"))          # range
for i in range(1,n+1):
    num = int(input("Enter value:"))
    data.append(num)
print("Data:",data)
```

```
Enter limit: 6
Enter value: 12
Enter value: 15
Enter value: 15
Enter value: 15
Enter value: 14
Enter value: 18

Data: [12, 15, 15, 15, 14, 18]
```

[54]:
```
# To find Factorial of all the value in the list:
data = [1,2,3,4,5,6,7,8,9]
print("Data:",data)
new = []
for i in data:
    f = 1
    for j in range(1,i+1):
        f = f*j
    new.append(f)
print("New:",new)
```

```
Data: [1, 2, 3, 4, 5, 6, 7, 8, 9]
New: [1, 2, 6, 24, 120, 720, 5040, 40320, 362880]
```

[55]:
```
# To print prime number in the list:
# create two different list of prime number and not prime number:
data = [12, 56, 4, 456, 61, 46, 156, 56, 23]
pr = []
```

```
npr = []
for i in data:
    flag = 0
    for j in range(2,i//2+1):
        if i%j==0:
            flag = 1
    if flag == 0:
        pr.append(i)
    else:
        npr.append(i)
print("Prime Number:",pr)
print("Not prime number:",npr)
```

```
Prime Number: [61, 23]
Not prime number: [12, 56, 4, 456, 46, 156, 56]
```

[56]:
```
# Make two different list of pallindrom number or not pallindrom numbers:
data = [121,123,343,345,5665,678,875,7843,984,789,989,12321]
pr = []
npr = []
for i in data:
    n = i
    sm = 0
    while n>0:
        r = n%10
        n = n//10
        sm = sm*10+r
    if sm == i:
        pr.append(i)
    else:
        npr.append(i)
print("Palindrom:",pr)
print("Not palindrm:",npr)
```

```
Palindrom: [121, 343, 5665, 989, 12321]
Not palindrm: [123, 345, 678, 875, 7843, 984, 789]
```

[57]:
```
# List Comprehension:
data = [1,2,3,4,5]
new = [i*i*i for i in data]
print(new)
```

```
[1, 8, 27, 64, 125]
```

[58]:
```
# Nested list: List within list
# 2D Array: collection of 1D Array
# 3D Array: collection of 2D Array
```

[59]:
```
data = [[353,609,805],
        [258,121,123],
        [153,343,345],
        [565,678,875],
        [743,984,789],
        [989,121,346]]
print("data:",data[2])
```

```
data: [153, 343, 345]
```

[60]:
```
data = [[353,609,805],
        [258,121,123],
        [153,343,345],
        [565,678,875],
        [743,984,789],
        [989,121,346]]
for i in range(0,len(data)):
    print(data[i])
```

```
[353, 609, 805]
[258, 121, 123]
[153, 343, 345]
[565, 678, 875]
[743, 984, 789]
[989, 121, 346]
```

[ ]:
```

```

[61]:
```
data = [[353,609,805],
        [258,121,123],
        [153,343,345],
        [565,678,875],
        [743,984,789],
        [989,121,346]]
for i in range(0,len(data)):
    for j in range(0,len(data[i])):
        print(data[i][j],end=" ")
    print()
```

```
353 609 805
258 121 123
153 343 345
565 678 875
743 984 789
989 121 346
```

[62]:
```
# Without Range:
data = [[353,609,805],
        [258,121,123],
        [153,343,345],
        [565,678,875],
        [743,984,789],
        [989,121,346]]
for i in data:
    for j in i:
        print(j,end=" ")
    print()
```

```
353 609 805
258 121 123
153 343 345
565 678 875
743 984 789
989 121 346
```

[63]:
```
# 3D Array:
data = [[[353,609,805],
        [258,121,123],
        [153,343,345]],
        [[565,678,875],
        [743,984,789],
        [989,121,346]],
        [[125,256,354],
        [125,256,351],
        [651,357,951]]]
for i in data:
        for j in i:
                for k in j:
                        print(k,end=' ')
                print()
        print()
```

```
353 609 805
258 121 123
153 343 345

565 678 875
743 984 789
989 121 346

125 256 354
125 256 351
651 357 951
```

[64]:
```
data = [[[353,609,805],
        [258,121,123],
        [153,343,345]],
        [[565,678,875],
        [743,984,789],
        [989,121,346]],
        [[125,256,354],
        [125,256,351],
        [651,357,951]]]
for i in range(0,len(data)):
        for j in range(0,len(data[i])):
                for k in range(0,len(data[i][j])):
                        print(data[i][j][k],end=" ")
                print()
        print("------------")
```

```
353 609 805
258 121 123
153 343 345
------------
565 678 875
743 984 789
989 121 346
------------
125 256 354
125 256 351
651 357 951
------------
```

[65]:
```
# Copy Function:
data = [1,2,3,4,5,6]
new = data.copy()        # new= list(data)
print(data)
print(new)
new[2]= 100
print(data)
print(new)
```

```
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 4, 5, 6]
[1, 2, 100, 4, 5, 6]
```

[66]:
```
# Extend Function:
data = [1,2,3,4,5,6]
new = [456,346,453,345,634,364,354]
data.extend(new)
```

```
print("With Extend",data)
print(new)
```

```
With Extend [1, 2, 3, 4, 5, 6, 456, 346, 453, 345, 634, 364, 354]
[456, 346, 453, 345, 634, 364, 354]
```

[67]:
```
data.append(new)
print("With append:",data)
```

```
With append: [1, 2, 3, 4, 5, 6, 456, 346, 453, 345, 634, 364, 354, [456, 346,
453, 345, 634, 364, 354]]
```

[68]:
```
# Tuple : Immutable, Ordered, ()
# Slicing operators can be used in tuples
```

[69]:
```
# Slicing Operators
data = (1,2,4,5,2,23,42,44,453)
print(data[0:8])
print(data[-1::-1])
print(data[0:])
print(data[:-1])
```

```
(1, 2, 4, 5, 2, 23, 42, 44)
(453, 44, 42, 23, 2, 5, 4, 2, 1)
(1, 2, 4, 5, 2, 23, 42, 44, 453)
(1, 2, 4, 5, 2, 23, 42, 44)
```

[70]:
```
# Some Tuple functions:
data = (1, 2, 4, 5, 2, 23, 42, 44, 453)
print(len(data))
print(min(data))
print(max(data))
print(data.count(2))
```

```
9
1
453
2
```

[71]:
```
# Nested Tuples:
```

[72]:
```
# it can be:
data = ([12,15,18],[35,32,31],[25,23,24])
data[0][2] = 16
print(data)
```

```
([12, 15, 16], [35, 32, 31], [25, 23, 24])
```

[73]:
```
# Also it can be
data = [(1,2,3),(4,5,6),(7,8,9)]
data[0]=(12,13)
print(data)
```

```
[(12, 13), (4, 5, 6), (7, 8, 9)]
```

[74]:
```
# If we want to covert a list into tuple:
data = (1,2,3,4)
data = list(data)
print(data)
```

```
[1, 2, 3, 4]
```

[75]:
```
# Dictionary : mutable , unordered , {} , keys and values : to handle␣
↪structured data
# keys are non-changeable, values are changeable
# keys: int, float, tuple, string
```

[76]:
```
data = {"Name":"Manish","Age":20,"City":"Dehradun"}
print(data)
print(data["Age"])
```

```
{'Name': 'Manish', 'Age': 20, 'City': 'Dehradun'}
20
```

[77]:
```
print(data.keys())
print(list(data.keys()))
print(list(data.keys())[1])
```

```
dict_keys(['Name', 'Age', 'City'])
['Name', 'Age', 'City']
Age
```

[78]:
```
print(data.values())
print(list(data.values()))
print(list(data.values())[1])
```

```
dict_values(['Manish', 20, 'Dehradun'])
['Manish', 20, 'Dehradun']
20
```

[79]:
```
print(list(data.items()))                # in pure list form:
print(list(data.items())[2][1])          # to print dehradun
```

```
[('Name', 'Manish'), ('Age', 20), ('City', 'Dehradun')]
Dehradun
```

```
[80]: for i,j in data.items():
          print(i," => ",j)
```

```
Name  =>  Manish
Age  =>  20
City  =>  Dehradun
```

```
[81]: Brothers = [{"Name":"Manish","Age":20,"Address":{"City":"Dehradun","Pin":
       ↪246482,"State":"Uk"},
                                    "Hobbies":["Reading","Cricket","Music"]},
                  {"Name":"Mohit","Age":19,"Address":{"City":"Shimla","Pin":
       ↪220456,"State":"HP"},
                                    "Hobbies":["Music","Vollyball","Gaming"]},
                  {"Name":"Priyanshu","Age":29,"Address":{"City":"Delhi","Pin":
       ↪200254,"State":"Dl"},
                                    "Hobbies":["Gaming","Cricket","Music"]}]
      print(Brothers[1])
      print(Brothers[1]["Address"])
      print(Brothers[1]["Address"]["City"])
      print(Brothers[1]["Hobbies"])
      print(Brothers[1]["Hobbies"][1])
```

```
{'Name': 'Mohit', 'Age': 19, 'Address': {'City': 'Shimla', 'Pin': 220456,
'State': 'HP'}, 'Hobbies': ['Music', 'Vollyball', 'Gaming']}
{'City': 'Shimla', 'Pin': 220456, 'State': 'HP'}
Shimla
['Music', 'Vollyball', 'Gaming']
Vollyball
```

```
[82]: Students = [{"Name":"Manish","Age":20},{"Name":"Mohit","Age":19,},{"Name":
       ↪"Priyanshu","Age":19,}]
      for i in Students:
          for a,b in i.items():
              print(a," => ",b)
```

```
Name  =>  Manish
Age  =>  20
Name  =>  Mohit
Age  =>  19
Name  =>  Priyanshu
Age  =>  19
```

```
[83]: data = {"Name":"Manish","Age":20}                # to change the value
      data["Age"]=21
      print(data)
```

```
{'Name': 'Manish', 'Age': 21}
```

```
[84]: # TO add the Keys and value
      data={"Name":"Manish","Age":20}
      data["Address"]="Dehradun"
      print(data)
```

```
{'Name': 'Manish', 'Age': 20, 'Address': 'Dehradun'}
```

```
[85]: # To delete the inserted pair:
      data={"Name":"Manish","Age":20}
      data.pop("Age")
      print(data)
```

```
{'Name': 'Manish'}
```

```
[86]: # SET::
      # Set: mutable, unordered, {}, it can not store duplicate values
```

```
[87]: data = {1,2,3,4,5,4,3,2}
      print(data)
```

```
{1, 2, 3, 4, 5}
```

```
[12]: data = {2,5,1,3,4,8}
      x = list(data)
      print(x)
      x.append(11)
      print(x)
      data = set(x)
      print(data)
```

```
[1, 2, 3, 4, 5, 8]
[1, 2, 3, 4, 5, 8, 11]
{1, 2, 3, 4, 5, 8, 11}
```

```
[88]: a = {1,2,3,4,5,6,7}
      b = {5,6,7,8,9,10}

      print(a.union(b))
      print(a.intersection(b))
      print(a.difference(b))
      print(b.difference(a))
      print(b.symmetric_difference(a))
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
{5, 6, 7}
{1, 2, 3, 4}
{8, 9, 10}
{1, 2, 3, 4, 8, 9, 10}
```

[89]:
```
# Function: block of code, set of instructions that is assigned for a␣
  ↪particular task
# Pre-defined and user defined
# pre-defined: print,input,int,sum,count,max,min
```

[90]:
```
def msg():                                          # Definition of the functions
    print("Hello world")
    print("Thank you")
msg()                                               # Calling of the functions
```

```
Hello world
Thank you
```

[91]:
```
def add():
    a = int(input("Enter a number:"))
    b = int(input("Enter a number:"))
    c = a+b
    print("Addition is =",c)
add()
```

```
Enter a number: 12
Enter a number: 18

Addition is = 30
```

[92]:
```
def fact():
    n = int(input("enter a numbr:"))
    f = 1
    for i in range(1,n+1):
        f = f*i
    print("Factorial is = ",f)
fact()
```

```
enter a numbr: 5

Factorial is =  120
```

[93]:
```
for i in range(1,6):                    # IF we want many times
    fact()
```

```
enter a numbr: 1

Factorial is =  1

enter a numbr: 2

Factorial is =  2

enter a numbr: 3

Factorial is =  6
```

```
enter a numbr: 45

Factorial is =  119622220865480194561963161495657715064383733760000000000

enter a numbr: 6

Factorial is =  720
```

[94]:
```python
def rev():
    n = int(input("Enter a number:"))
    sm = 0
    while n>0:
        r = n%10
        n = n//10
        sm = sm*10+r
    print("Reverse:",sm)
rev()
```

```
Enter a number: 7

Reverse: 7
```

[95]:
```python
# Parameterized Function:
def add(a,b):                                      # Parameterized Function
    c = a+b
    print("Addition is =",c)
add(45,56)                                         # Arguments
add(5,6)


x = int(input("Enter a number:"))
y = int(input("Enter a number:"))
add(x,y)
```

```
Addition is = 101
Addition is = 11

Enter a number: 12
Enter a number: 18

Addition is = 30
```

[96]:
```python
# With print statement :
def add(a,b):                              # Parameterized Function
    c = a+b
    return c                               # Returning
print(add(12,18))
```

```
30
```

[97]:
```
# Making reverse program with parameterized function:
def rev(n):
    sm = 0
    while n>0:
        r = n%10
        n = n//10
        sm = sm*10+r
    return sm
print(rev(135))
```

```
531
```

[98]:
```
# TASK:-

# 1: No return, No Perameterized
# 2: No return, Parameterized
# 3: Returning, No Perameterized
# 4: Returning, Perameterized
```

[99]:
```
# Prime function program with 4rth condition:
def prime(n):
    flag = 0
    for i in range(2,n//2+1):
        if n%i==0:
            flag = 0
            break
    if flag == 0:
        return "Prime Number"
    else:
        return "Not prime number"
print(prime(23))
```

```
Prime Number
```

[100]:
```
# Factorial function with 3rd condition:
def fact():
    n = 4
    f = 1
    for i in range(1,n+1):
        f = f*i
    return f
print(fact())
```

```
24
```

[106]:
```
# Lcm program with 1 st condition:
def lcm():
    a = 12
    b = 15
    mx = a if a>b else b
    for i in range(mx,a*b+1):
        if i%a==0 and i%b==0:
            print("LCM:",i)
            break
lcm()
```

```
LCM: 60
```

[107]:
```
# Advantages of the function:
# 1: reduce line of code
# 2: Easy to understand and Manage
# 3: Efficient Memory Use (Reduce Memory Consumption)

# Disadvantage : increases time cunsumption

# time complexity
# space complexity
```

[108]:
```
# Types of Argument:
# 1: Positional Arg
# 2: Keyword Arg
# 3: Default Arg
# 4: Variable length Arg
```

[110]:
```
def emp(eid,name,sal):
    print("ID:",eid)
    print("Name:",name)
    print("Salary:",sal)
emp(111,"Mohit",35000)                    # Positional
emp(sal=45000,eid=112,name="Mayank")      # Keyword
```

```
ID: 111
Name: Mohit
Salary: 35000
ID: 112
Name: Mayank
Salary: 45000
```

[112]:
```
# Default Argument:
def add(a=0,b=0,c=0):                    # Default argument
    return a+b+c
print(add(23,27))
```

```
50
```

[113]:
```
# Variable length Arg:
def add(*args):                         # variable length arg
    sm = 0
    for i in range(0,len(args)):
        sm = sm+args[i]
    return sm
res = add(23,27,10,15,25)
print(res)
```

```
100
```

[114]:
```
def emp(**kwargs):
    for i,j in kwargs.items():
        print(i," = > ",j)
emp(sal=45000,eid=111,name="mohit",address="Shimla",posr="clerk")              #␣
 ↪Keywords Arg
```

```
sal  = >  45000
eid  = >  111
name  = >  mohit
address  = >  Shimla
posr  = >  clerk
```

[115]:
```
# LCM Program when we have to pass the values:
def lcm(*data):
    mx = max(data)
    ml = 1
    for i in data:
        ml= ml*i
    for i in range(mx,ml+1):
        count= 0
        for j in data:
            if i%j == 0:
                count = count+1
        if count==len(data):
            return i
res=lcm(12,15,20)
print("LCM is",res)
```

```
LCM is 60
```

[128]:
```
# Anonymous function, linear function, inline function
# Lambda: a function can be stored in var, can be passed as an argument, can be␣
 ↪returned as a value
```

[129]:
```
# Lambda:
cube = lambda x:x*x*x
```

```
print(cube(5))
# OR
print((lambda x:x**3)(5))
```

```
125
125
```

[116]:
```
# Recursion: a function that calls itself:
# Fibonacci series:
```

[9]:
```
# Recursion:
'''
def vip():
    print("Hello world!")
    vip()
vip()

'''
```

[9]:
```
'\ndef vip():\n    print("Hello world!")\n    vip()\nvip()\n\n'
```

[131]:
```
# Factorial function with recursion (with conditions):
def fact(n):
    if n==1:
        return 1
    else:
        return n*fact(n-1)
res = fact(5)
print(res)
```

```
120
```

[132]:
```
# Fibonacci series:
# Without Recursion:

def fib(n):
    first = 0
    second = 1
    print(first)
    print(second)
    for i in range(1,n-1):
        third = first + second
        print(third)
        first = second
        second = third
fib(10)
```

```
0
```

```
1
1
2
3
5
8
13
21
34
```

[135]:
```python
def fib(n):
    a = 0
    b = 1
    count = 0
    while True:
        print(a)
        a,b=b,a+b
        count = count+1
        if count == n:
            break

fib(10)
```

```
0
1
1
2
3
5
8
13
21
34
```

[136]:
```python
def fib(n):
    a = 0
    b = 1
    count = 0
    for i in range(1,n+1):
        print(a)
        a,b=b,a+b
fib(10)
```

```
0
1
1
2
3
```

```
5
8
13
21
34
```

[ ]:

```

```

[118]:

```
def fib(n):
    if n<=1:
        return n
    else:
        return fib(n-1)+fib(n-2)
for i in range(0,10):
    print(fib(i),end=" ")
```

```
0 1 1 2 3 5 8 13 21 34
```

[124]:

```
# Another Way:
def fib(n):
    if n <= 0:
        return "Input should be a positive integer."
    elif n == 1:
        return 0
    elif n == 2:
        return 1
    else:
        return fib(n-1) + fib(n-2)
n = 10
for i in range(1,n+1):
    print(fib(i),end=" ")
```

```
0 1 1 2 3 5 8 13 21 34
```

[164]:

```
# Make a program of Prime numbers which should be returning also with the use␣
 ↪of recursion:
def prime(*args):
    pr = []
    for i in args:
        flag = 0
        for j in range(2,i//2+1):
            if i % j == 0:
                flag = 1
                break
        if flag == 0:
            pr.append(i)
    return pr
res = prime(1,2,3,4,5,6,7,8,9,23,34,55,35,89,34,77,33,93)
```

```
print(res)
```

```
[1, 2, 3, 5, 7, 23, 89]
```

[126]:
```
# map, reduce, filter (HOF: Higher Order Functinon)

# Map:
```

[127]:
```
data = [1,2,3,4,5,6,7]
def cube(n):
    return n*n*n
print(list(map(cube,data)))
```

```
[1, 8, 27, 64, 125, 216, 343]
```

[5]:
```
# Create a Python program that uses the map() function to convert a list of␣
 ↪names to uppercase:
data = ["mohit","aman","manish"]
print(list(map(lambda x:x.upper(),data)))
```

```
['MOHIT', 'AMAN', 'MANISH']
```

[138]:
```
# With Lambda-
data = [1,2,3,4,5,6,7]
print(list(map(lambda x:x*x*x,data)))
```

```
[1, 8, 27, 64, 125, 216, 343]
```

[139]:
```
# func to make all string capital:
data = ["Singh","Vikas","Arjun","Vijay","Shivam"]
print(list(map(lambda x:x.upper(),data)))
```

```
['SINGH', 'VIKAS', 'ARJUN', 'VIJAY', 'SHIVAM']
```

[140]:
```
# IF we want first letter also in capital:
print(list(map(lambda x:x[0].upper(),data)))
```

```
['S', 'V', 'A', 'V', 'S']
```

[141]:
```
# Filter
```

[142]:
```
# Where name starts form A:
data = ["Arun","Vikas","arjun","Vijay","Ankit"]
print(list(filter(lambda x:x[0].upper()=="A",data)))
```

```
['Arun', 'arjun', 'Ankit']
```

[143]:
```
# Even numbers:
new=[1,2,3,4,5,6,7,8,9,45,16,45,94,64,46,59,46]
print(list(filter(lambda x:x%2==0,new)))
```

```
[2, 4, 6, 8, 16, 94, 64, 46, 46]
```

[144]:
```
# Reduce
```

[145]:
```
from functools import reduce
data = ["Arun","Vikas","arjun","Vijay","Ankit"]
new=[1,2,3,4,5,6,7,8,9,45,16,45,94,64,46,59,46]
# for sum:
print(reduce(lambda x,y:x+y,new))
# for concate:
print(reduce(lambda x,y:x+" "+y,data))
```

```
460
Arun Vikas arjun Vijay Ankit
```

[146]:
```
# Modules: every file of python is a module:
# Module is a file that contains function and classes and that can be import in␣
 ↪another file

# import fact as f
# from fact import fact
# from fact import *
```

[148]:
```
import math as m
print(m.sqrt(25))
print(m.gcd(10,15))
print(m.lcm(10,15))
print(m.cos(45))
print(m.pow(10,2))
```

```
5.0
5
30
0.5253219888177297
100.0
```

[162]:
```
# Calendar module:
import calendar
print(calendar.month(2020,4))
print(calendar.calendar(2020,2,1,6))
```

```
     April 2020
Mo Tu We Th Fr Sa Su
       1  2  3  4  5
```

```
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

                                  2020

      January                   February                   March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
       1  2  3  4  5                      1  2                         1
 6  7  8  9 10 11 12       3  4  5  6  7  8  9       2  3  4  5  6  7  8
13 14 15 16 17 18 19      10 11 12 13 14 15 16       9 10 11 12 13 14 15
20 21 22 23 24 25 26      17 18 19 20 21 22 23      16 17 18 19 20 21 22
27 28 29 30 31            24 25 26 27 28 29         23 24 25 26 27 28 29
                                                    30 31

       April                      May                       June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
       1  2  3  4  5                   1  2  3       1  2  3  4  5  6  7
 6  7  8  9 10 11 12       4  5  6  7  8  9 10       8  9 10 11 12 13 14
13 14 15 16 17 18 19      11 12 13 14 15 16 17      15 16 17 18 19 20 21
20 21 22 23 24 25 26      18 19 20 21 22 23 24      22 23 24 25 26 27 28
27 28 29 30               25 26 27 28 29 30 31      29 30

        July                     August                  September
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
       1  2  3  4  5                      1  2          1  2  3  4  5  6
 6  7  8  9 10 11 12       3  4  5  6  7  8  9       7  8  9 10 11 12 13
13 14 15 16 17 18 19      10 11 12 13 14 15 16      14 15 16 17 18 19 20
20 21 22 23 24 25 26      17 18 19 20 21 22 23      21 22 23 24 25 26 27
27 28 29 30 31            24 25 26 27 28 29 30      28 29 30
                          31

      October                   November                  December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
          1  2  3  4                         1          1  2  3  4  5  6
 5  6  7  8  9 10 11       2  3  4  5  6  7  8       7  8  9 10 11 12 13
12 13 14 15 16 17 18       9 10 11 12 13 14 15      14 15 16 17 18 19 20
19 20 21 22 23 24 25      16 17 18 19 20 21 22      21 22 23 24 25 26 27
26 27 28 29 30 31         23 24 25 26 27 28 29      28 29 30 31
                          30
```

[163]:
```python
# OR
for i in range(1,13):
    print(calendar.month(2020,i))
```

```
    January 2020
```

```
Mo Tu We Th Fr Sa Su
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

   February 2020
Mo Tu We Th Fr Sa Su
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29

     March 2020
Mo Tu We Th Fr Sa Su
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31

     April 2020
Mo Tu We Th Fr Sa Su
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

      May 2020
Mo Tu We Th Fr Sa Su
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

     June 2020
Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
```

```
     July 2020
Mo Tu We Th Fr Sa Su
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

    August 2020
Mo Tu We Th Fr Sa Su
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31

   September 2020
Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30

    October 2020
Mo Tu We Th Fr Sa Su
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

   November 2020
Mo Tu We Th Fr Sa Su
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30

   December 2020
Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
```

```
28 29 30 31
```

[151]:
```
# Datetime module:
import datetime
x = datetime.datetime.now()
print(x)                          # current date and time
print(x.year)                     # Year
print(x.strftime("%B"))           # month
print(x.strftime("%A"))           # day of week
```

```
2024-09-28 12:42:41.773177
2024
September
Saturday
```

[152]:
```
# Local and global:
a = 100               # Global Variable
def msg():
    a = 10            # Local variable
    print(a)
msg()
print(a)

a = 100               # Global Variable
def msg(m):
    global a          # if we don'nt pass this statement then the output will be␣
 ↪100 cuz 100 is global variable
    if m =="Hi":
        a = 101       # local variable
    else:
        a = 102
msg("Hi")
print(a)
```

```
10
100
101
```

[153]:
```
# Exception Handling: exception => error or we can say run time errors are␣
 ↪exceptions-

# - compile time error: syntax error,
# - run time error: logical error, bug, exceptions
```

[154]:
```
a = 5
b = "Hello"
```

```
try:                          # If the code is right and there is no error
    c = a/b
    print("result is = ",c)
except ZeroDivisionError: # If we want to simpalise the error that anyone can␣
 ↪understand
    print("You are trying to divide a number by zero")
except TypeError:
    print("You are using different types of values in division operation")
except NameError:
    print("You are using undefined variable")
except:                       # If we are not finding the errors
    print("undefined Error Occured")
finally:                      # if we want these msg even there is an error
    print("thank you")
    print("visit again")
```

```
You are using different types of values in division operation
thank you
visit again
```

[156]:
```
# File Handling:

file = open("abcd.txt","w") # "w" for overwrite
#file = open("abcd.txt","a")   # "a" for write at the end
data = input("Enter your text:")
file.write(data)
file.close()
print("task Done, File Created and updated")



file = open("abcd.txt","r")           # open file1
data = file.read()                    # reading from file 1
file2 = open("xyz.txt","w")           # open file 2
file2.write(data)                     # writing to file 2
print(data)                           # output screen
file.close()
file2.close()
print("task Done, file Created and Updated")
```

```
Enter your text: Hi my name is manish bisht.
```

```
task Done, File Created and updated
Hi my name is manish bisht.
task Done, file Created and Updated
```

[ ]:
```
# OOPS Concepts : Object Oriented Programming System-
```

[157]:
```python
class student:
    def st_info(self,rn,n,con):
        print("Roll No:",rn)
        print("Name:",n)
        print("Contact:",con)
    def st_add(self,c,p,s):
        print("City:",c)
        print("Pin:",p)
        print("State:",s)

st1 = student()
st1.st_info(1,"Mohit",154346)
st1.st_add("Shimla",5846,"HP")

print(" ")

st2 = student()
st2.st_info(2,"Parveen",197946)
st2.st_add("Rampur",4653,"HP")
```

```
Roll No: 1
Name: Mohit
Contact: 154346
City: Shimla
Pin: 5846
State: HP

Roll No: 2
Name: Parveen
Contact: 197946
City: Rampur
Pin: 4653
State: HP
```

[158]:
```python
class student:
    def set_info(self,rn,n,con):
        self.rollno=rn
        self.name=n
        self.contact=con
    def get_info(self):
        print("Roll No : ",self.rollno)
        print("Name : ",self.name)
        print("Contact : ",self.contact)
    def set_add(self,c,p,s):
        self.city=c
        self.pin=p
        self.state=s
    def get_add(self):
        print("City : ",self.city)
        print("Pin : ",self.pin)
        print("State : ",self.state)
st1=student()
st1.set_info(1,"Mohit",154346)
st1.set_add("Shimla",5846,"HP")
st1.get_info()
st1.get_add()
```

```
Roll No :  1
Name :  Mohit
Contact :  154346
City :  Shimla
Pin :  5846
State :  HP
```

[160]:
```
class student:
    def __init__(self,rn,n,con,c,p,s):
        self.rollno=rn
        self.name=n
        self.contact=con
        self.city=c
        self.pin=p
        self.state=s
    def get_info(self):
        print("Roll No : ",self.rollno)
        print("Name : ",self.name)
        print("Contact : ",self.contact)
    def get_add(self):
        print("City : ",self.city)
        print("Pin : ",self.pin)
        print("State : ",self.state)
st1 = student(1,"Mohit",154464,"Shimla",265564,"HP")
print(st1.pin)
print(st1.contact)
```

```
265564
154464
```

[161]:
```
# Task: to make a employee class:

class employee:
    def __init__(self,n,a,c,e,p,s):
        self.name=n
        self.age=a
        self.city=c
        self.emp_id=e
        self.post=p
        self.salary=s
    def get_info(self):
        print("Name: ",self.name)
        print("Age: ",self.age)
        print("City: ",self.city)
    def get_salary(self):
        print("Employee ID: ",self.emp_id)
        print("Post: ",self.post)
        print("Salary: ",self.salary)
emp1 = employee("Mohit",19,"Shimla","E01","Clerk",35000)
print(emp1.city)
print(emp1.post)
```

```
Shimla
Clerk
```

[5]:
```
# import a random number then We have three chance to guess the number if we␣
 ↪guess right then print you win! else you lose!
import random
n = random.randint(1,10)
num1 = int(input("First Guess:"))
if n == num1:
    print("you win!")
else:
    num2 = int(input("Second Guess:"))
    if n==num2:
        print("you win!")
    else:
        num3 = int(input("Third Guess:"))
        if n==num3:
            print("you win!")
        else:
            print("you lose!")
print("the random number is ",n)
```

```
First Guess: 2
Second Guess: 5
Third Guess: 7

you lose!
the random number is  10
```